

- 1- Compléter le code de la méthode constructeur
- 2- Compléter le code de la méthode `__str__()`
- 3- Compléter le code de la méthode `setAge()`
- 4- Compléter le script de la méthode `getAge()`
- 5- En poo, quel nom qualificatif donne-t-on aux variables `nom`, `prenom`, `instrument`, `surnom` et `age` ?
- 6- En poo, quel nom qualificatif donne-t-on à la variable `nbr_musiciens` ?

Exercice 2. : Récursivité

On considère le programme suivant :

```
def f(a,b) :
    if b == 1 :
        return a
    return a + f(a,b-1)

val = f(7,9)
print(val)
```

- 1- Exécuter ci-dessous ce code « à la main »

- 2- Donner ci-dessous une version non récursive de ce même code :

```
def F(a,b) : ..
```

Exercice 3. : Récursivité

On considère le programme suivant :

```
def f(n) :
    if n < 2 :
        return 1
    else :
        return f(n-1) + f(n-2)

val = f(7)
print(val)
```

- ⇒ Exécuter ci-dessous ce code « à la main »

⇒ Se loguer avec l'identifiant : **exam02.eleve**

Mot de passe :

Le code à réaliser sera appelé `ds_mon_nom.py` . Il est à déposer en fin d'épreuves dans le répertoire : ***Examens(Z :)/exam02/copies/***

⇒ **Copier** le dossier *NSI-15nov2022* : ***Examens(Z :)/exam02/sujets/NSI-15nov2022*** sur le disque dur de l'ordinateur (sur le *bureau* ou dans le répertoire *Mes documents*) et le décompresser.

Exercice 4. : POO : classes *Musicien* et *Groupe*

⇒ Ouvrir le fichier `exercice4_annonce.py` . Le renommer avec le nom `ds_mon_nom.py`.

Sont définies dans ce fichier 2 classes. La classe *Musicien* étudiée dans la partie A et la classe *Groupe* qui permet de définir les musiciens qui appartiennent à un groupe musical.

⇒ Compléter le code de la classe *Groupe* et celui du dernier `print()` ci-dessous, afin que l'exécution des lignes suivantes :

```
#Main
g = Groupe("Orelsan")
g.setMembres("Cotentin","Aurélien","chant", "Orelsan",40)
g.setMembres("Le Carpentier","Matthieu","beatmaker", "Skread",41)
g.setMembres("Preau","Adam","beatmaker", "Phazz",31)
g.setMembres("Ardan","Edouard","guitariste" , "Eddie Purple")
g.setMembres("Dyens","Manu","batter")
print(g)
moy = g.moyenneAge()
print(moy)
b = Groupe("Beatles")
print(f"Il y a ..... instances de la classe Groupe créées")
```

donne précisément dans la console :

```
>>> (executing file "musique.py")
Composition du groupe d'Orelsan : 5 membres
- Orelsan, chant, de son vrai nom : Cotentin Aurélien
- Skread, beatmaker, de son vrai nom : Le Carpentier Matthieu
- Phazz, beatmaker, de son vrai nom : Preau Adam
- Eddie Purple, guitariste, de son vrai nom : Ardan Edouard
- Dyens Manu, batter

37.3
Il y a 2 instances de la classe Groupe créées
```

Exercice 5. : Récursivité

On définit la suite de nombre suivante : $3 \Rightarrow 6 \Rightarrow 12 \Rightarrow 24 \Rightarrow 48 \Rightarrow 96 \Rightarrow 192 \Rightarrow \dots$

Le premier de ces nombres est donc $u_1 = 3$. On a ensuite $u_2 = 6$, $u_3 = 12$, ..., $u_7 = 192$

On a globalement : $u_n = 2 u_{n-1}$ avec $u_1 = 3$.

Dans le fichier .py complété précédemment :

- 1- Coder une fonction *iter(n)* qui renvoie le $n^{\text{ième}}$ terme en utilisant un algorithme itératif
- 2- Coder une fonction *rec(n)* qui renvoie le $n^{\text{ième}}$ terme en utilisant un algorithme récursif

Exercice 6. : Récursivité

La fonction *compteur(l, val)* renvoie le nombre d'éléments de la liste numérique *l* qui sont égaux à la variable *val*.

L'exécution des lignes donnera dans la console :

```
nbr = compteur([1,7,-8,9,1,20],1)
print(nbr)
print(compteur([1,7,-8,9,1,20],40))
```

```
>>> (executing file "compteur.py")
2
0
```

Dans le fichier .py complété précédemment :

- 1- Coder une version itérative de cette fonction. On la nommera *compteurIT()*.

- 2- Coder une version récursive de cette fonction. On la nommera *compteur()*. On utilisera la méthode *pop()* qui supprime le dernier élément d'une liste et renvoie sa valeur. Par exemple :

```
>>> l = [1,7,-8,9,1,20]
>>> l.pop()
20
>>> l
[1, 7, -8, 9, 1]
```